An Image is Worth 16 x 16 Words:

Transformers for Image Classification at Scale

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and Jakob Uszkoreit

Google Brain, ICLR 2021

Presentation by Soham De*

*a mere 2nd year UG who is yet to take IML, so please forgive errors

Agenda

- ➔ **Overview and Results**
  This will be a rant on why I find this paper interesting
- ➔ **A History Lesson**
  A quick refreshers for pre-requisites
- ➔ **Vision Transformer (ViT)**
  A discussion on the architecture proposed by this paper
- ➔ **Discussion**
  An extended discussion on the implications and shortcomings

Overview and Results

Overview

“While the Transformer architecture has become the de-facto standard for NLP tasks, its applications to computer vision remain limited. Invision, attention is either applied in conjunction with convolutional networks, orused to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art CNNs while requiring substantially fewer computational resources to train”

Results

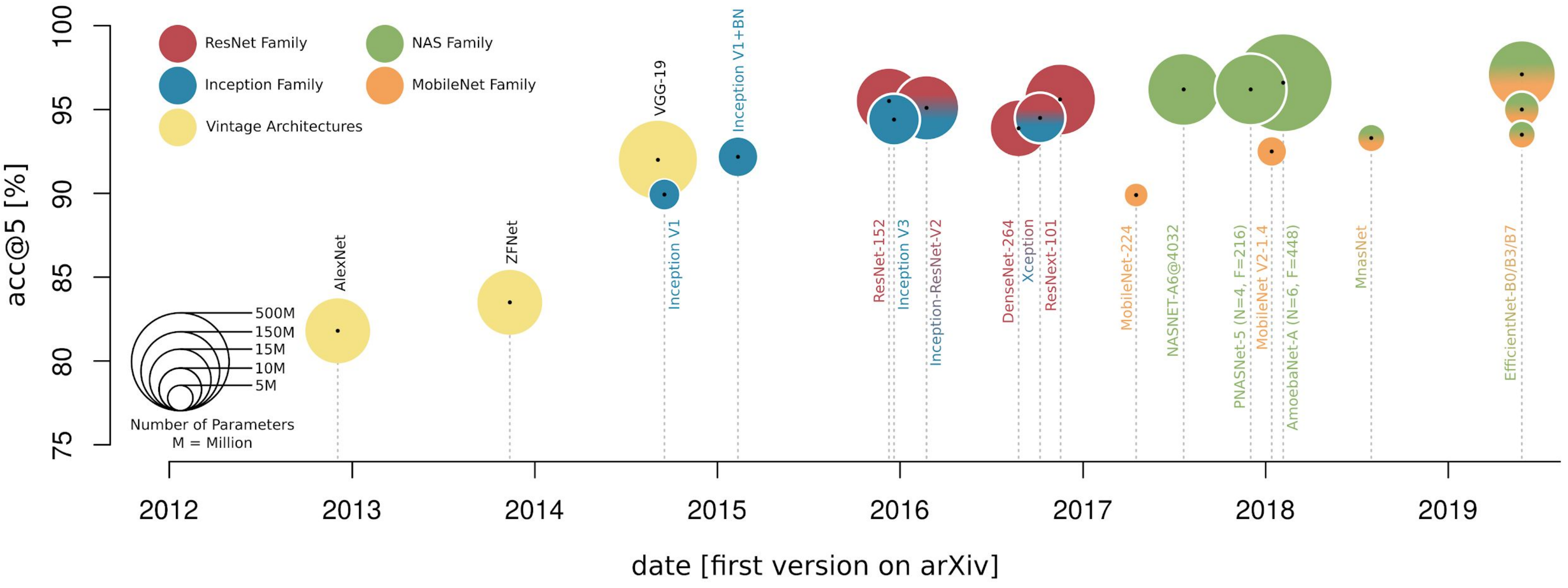
“We find that large scale training trumps inductive bias. Our Vision Transformer (ViT) attains excellent results when pre-trained at sufficient scale and transferred to tasks with fewer data points. When pre-trained on the public ImageNet-21k dataset or the in-house JFT-300M dataset, ViT approaches or beats state of the art on multiple image recognition benchmarks. In particular, the best model reaches the accuracy of 88.55% on ImageNet, 90.72% on ImageNet-ReaL, 94.55% on CIFAR-100, and 77.63% on the VTAB suite of 19 tasks.”

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21K (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** $\pm$ 0.04	87.76 $\pm$ 0.03	85.30 $\pm$ 0.02	87.54 $\pm$ 0.02	88.4/88.5*
ImageNet ReaL	**90.72** $\pm$ 0.05	90.54 $\pm$ 0.03	88.62 $\pm$ 0.05	90.54	90.55
CIFAR-10	**99.50** $\pm$ 0.06	99.42 $\pm$ 0.03	99.15 $\pm$ 0.03	99.37 $\pm$ 0.06	–
CIFAR-100	**94.55** $\pm$ 0.04	93.90 $\pm$ 0.05	93.25 $\pm$ 0.05	93.51 $\pm$ 0.08	–
Oxford-IIIT Pets	**97.56** $\pm$ 0.03	97.32 $\pm$ 0.11	94.67 $\pm$ 0.15	96.62 $\pm$ 0.23	–
Oxford Flowers-102	99.68 $\pm$ 0.02	**99.74** $\pm$ 0.00	99.61 $\pm$ 0.02	99.63 $\pm$ 0.03	–
VTAB (19 tasks)	**77.63** $\pm$ 0.23	76.28 $\pm$ 0.46	72.72 $\pm$ 0.21	76.29 $\pm$ 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

A History Lesson

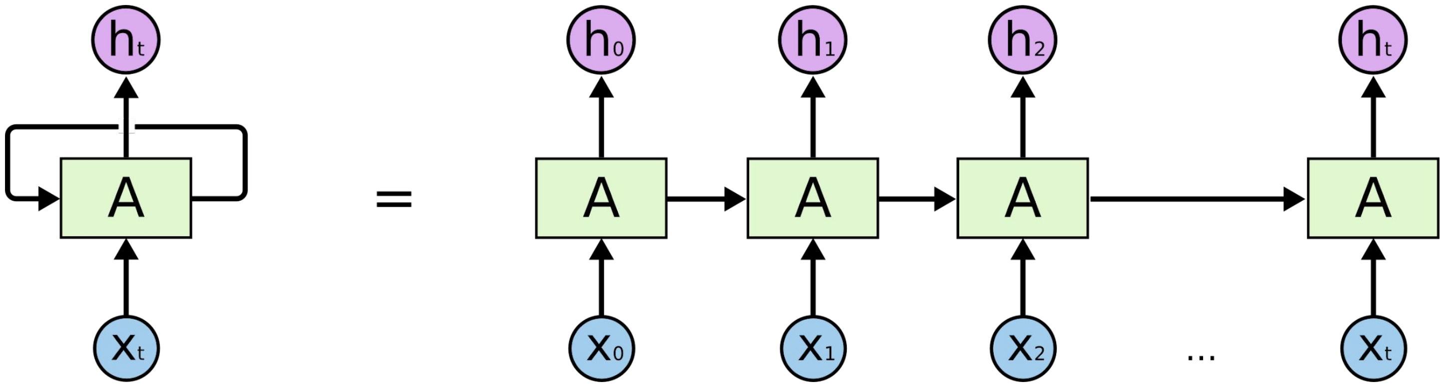
A History Lesson

CNNs vs IMAGENET (over the years)



A History Lesson

RNN (Hinton et al, 1986)

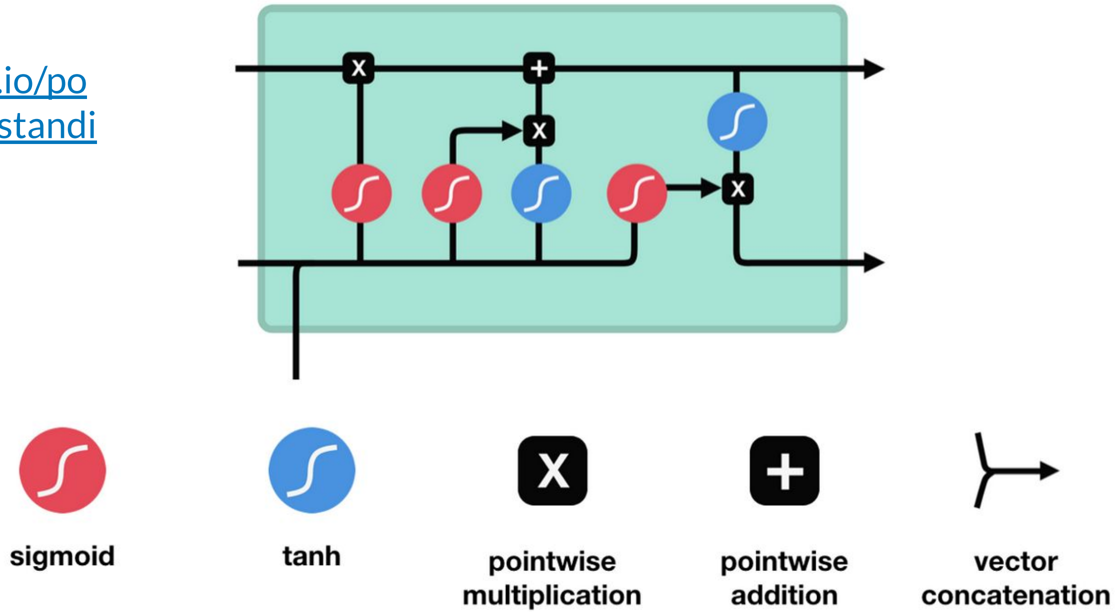


RNN

- Very deep layers
- Vanishing and Exploding Gradients
- Long Term Dependency issues

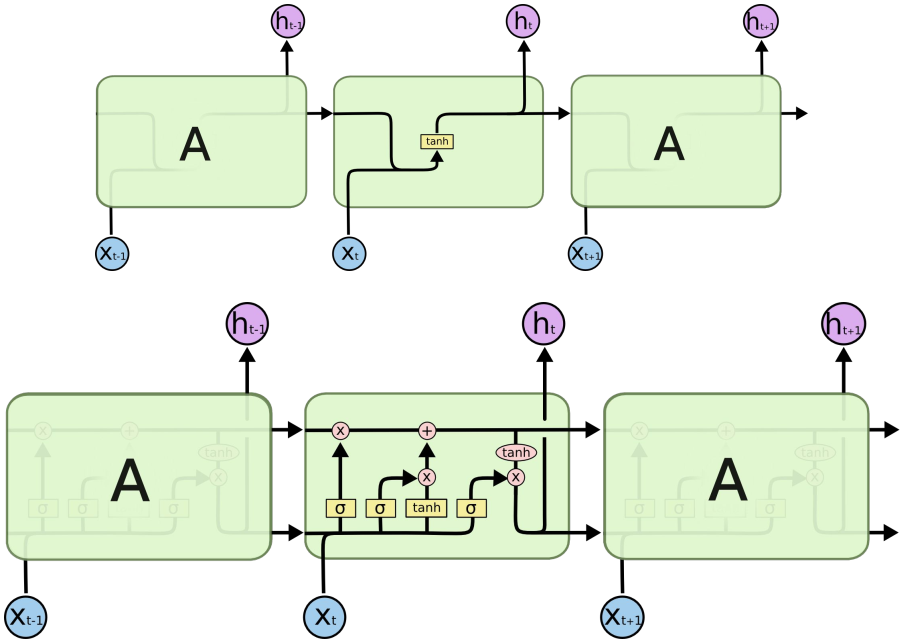
LSTM (Hochreiter & Schmidhuber, 1997)

https://colah.github.io/posts/2015-08-Understanding-LSTMs/



LSTM

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

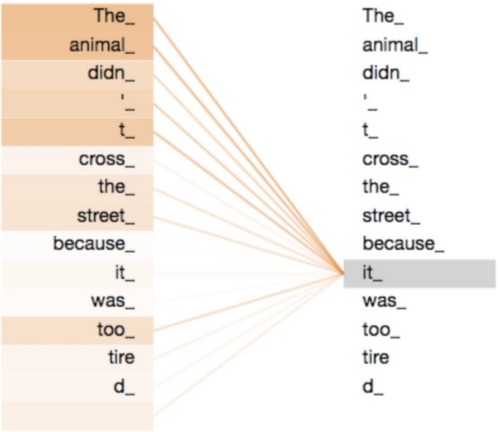


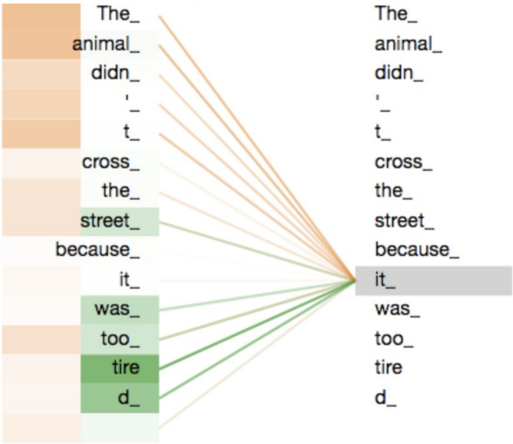
LSTM

- Solves vanishing gradients problem
- More computationally expensive (slower)
- Not parallelizable
- Transfer Learning didn’t really work on these

Attention (Xu et al, 2015)

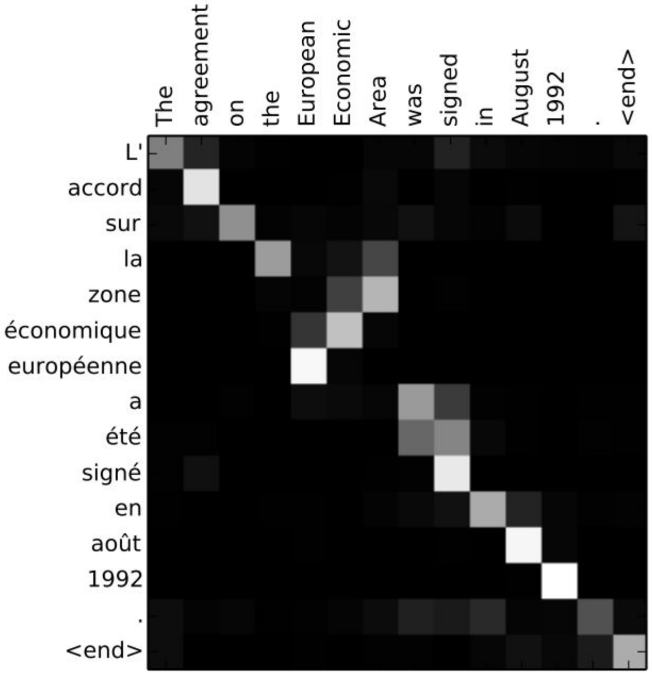
"The animal didn't cross the street because it was too tired"





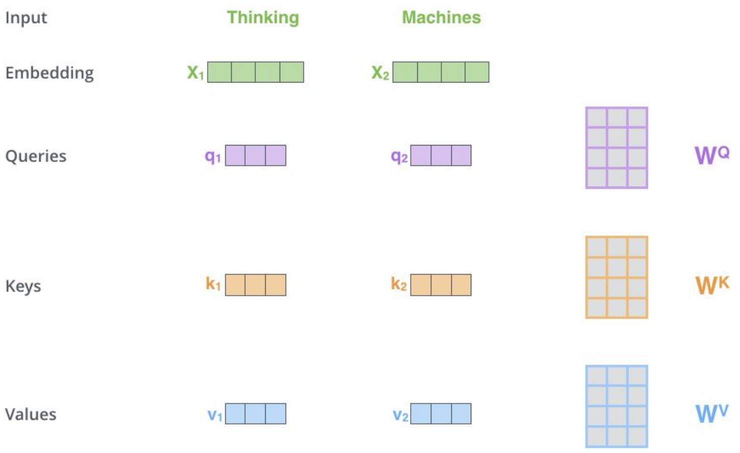
A History Lesson

Attention (in NMT)

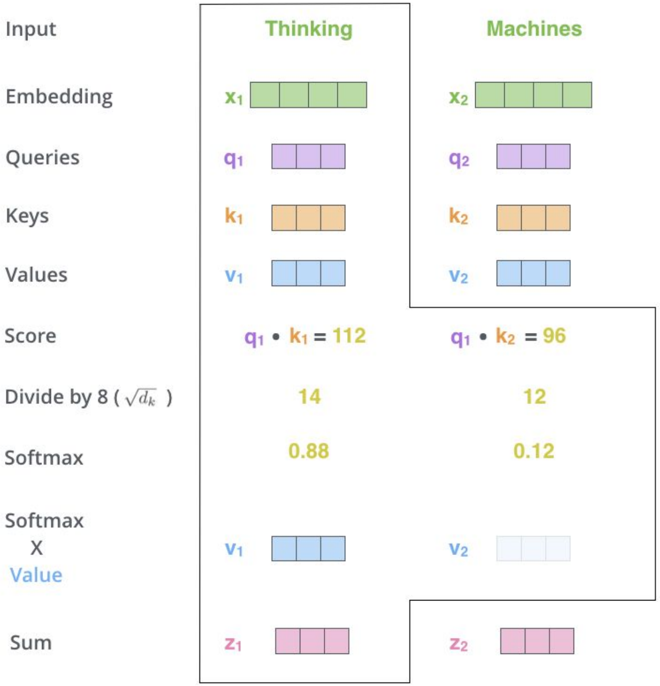


A History Lesson

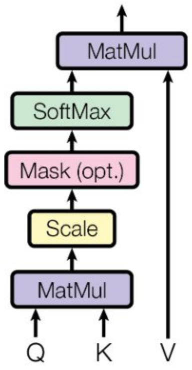
Attention

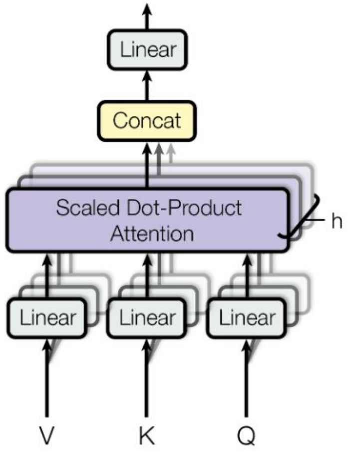


$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



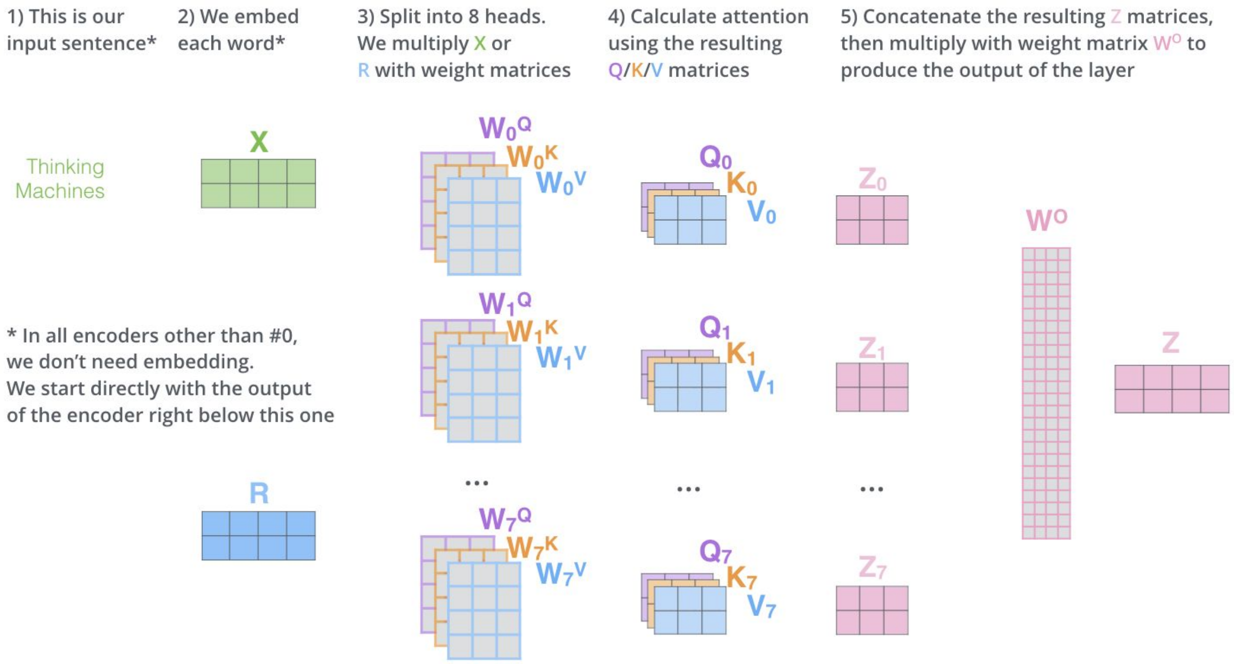
Attention





$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Attention



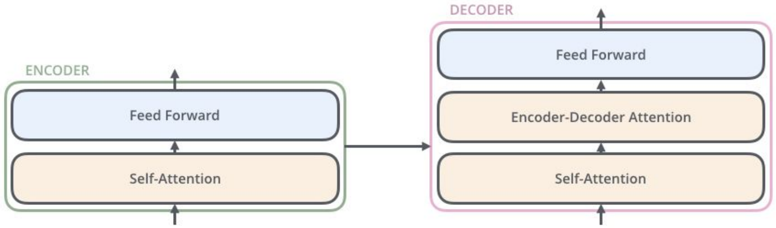
1) This is our input sentence*
2) We embed each word*
3) Split into 8 heads. We multiply X or R with weight matrices
4) Calculate attention using the resulting Q/K/V matrices
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer
Thinking Machines
X
R
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one
W_0^Q W_0^K W_0^V
W_1^Q W_1^K W_1^V
...
W_7^Q W_7^K W_7^V
Q_0 K_0 V_0
Q_1 K_1 V_1
...
Q_7 K_7 V_7
Z_0
Z_1
...
Z_7
W^O
Z

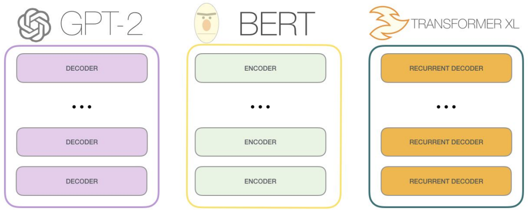
Vision Transformer

Transformer (Vaswani et al, 2017)

http://jalammar.github.io/illustrated-transformer/

http://nlp.seas.harvard.edu/2018/04/03/attention.html





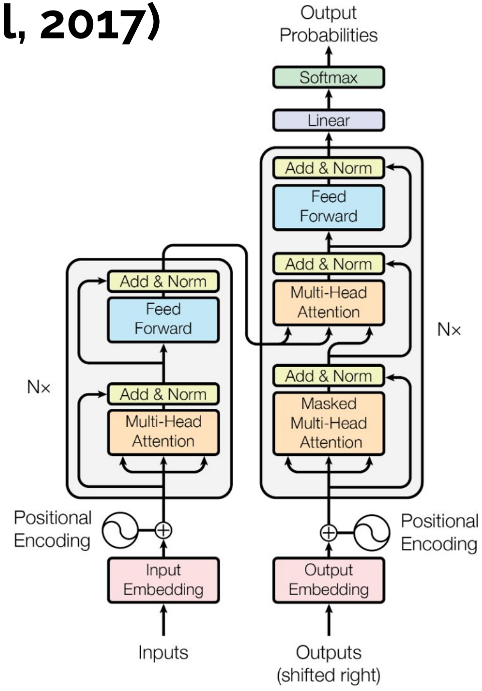
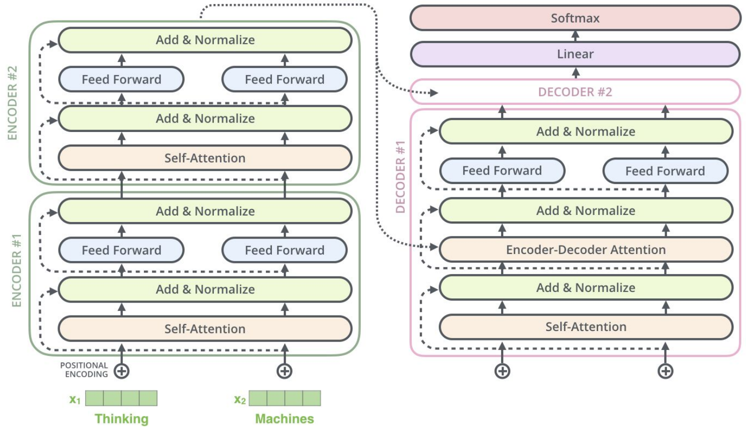
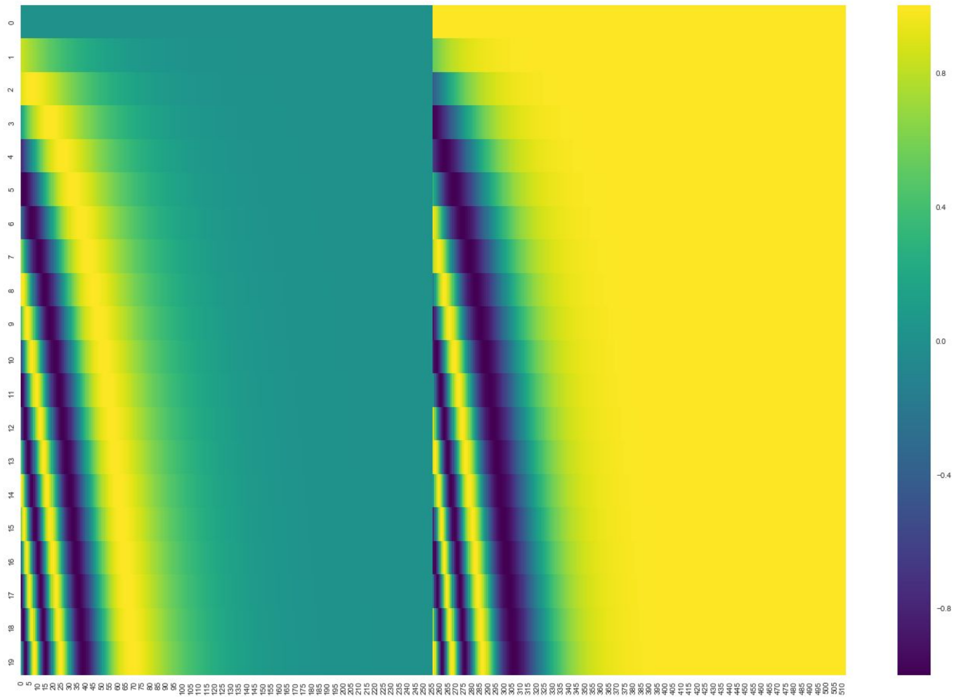


Figure 1: The Transformer - model architecture.

Transformer

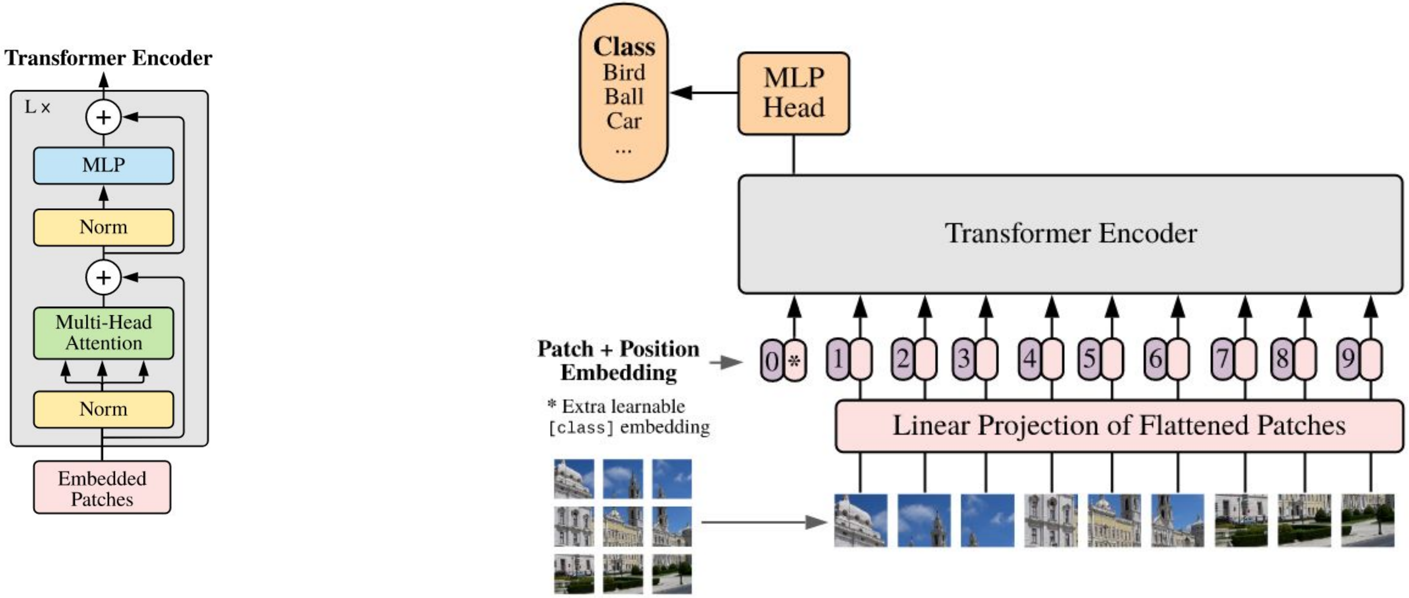




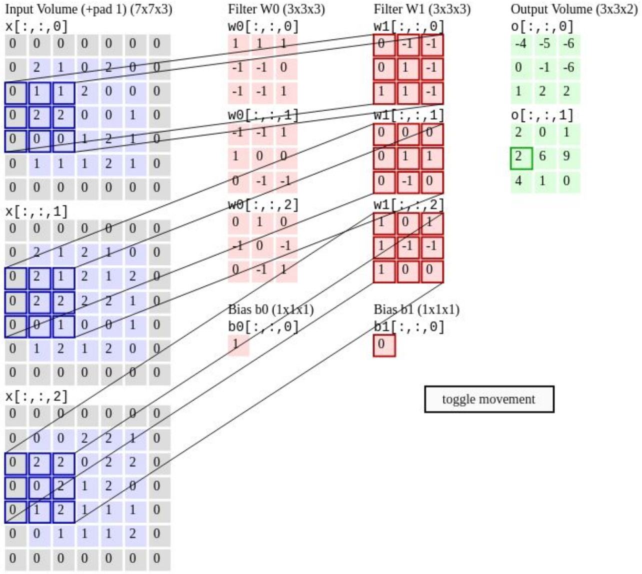
Transformer

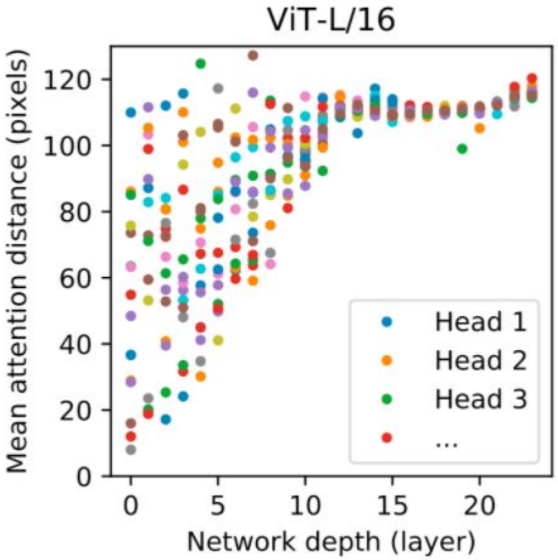
- Solves vanishing gradients problem
- Parallelizable; hence faster, easier to train
- Entire sequence at once (positional embeddings ftw)
- Transfer Learning works!
- Attention is O(N^2) 😔

Vision Transformer (Dosovitskiy et al, 2021)



ViT vs CNNs


Input Volume (+pad 1) (7x7x3)
x[:,:,0]
x[:,:,1]
x[:,:,2]
Filter W0 (3x3x3)
w0[:,:,0]
w0[:,:,1]
w0[:,:,2]
Bias b0 (1x1x1)
b0[:,:,0]
Filter W1 (3x3x3)
w1[:,:,0]
w1[:,:,1]
w1[:,:,2]
Bias b1 (1x1x1)
b1[:,:,0]
Output Volume (3x3x2)
o[:,:,0]
o[:,:,1]
toggle movement


ViT-L/16
Mean attention distance (pixels)
Network depth (layer)
Head 1
Head 2
Head 3
...

Discussion

- ➔ So is this the end of CNNs?
- ➔ Green AI?
- ➔ Are Double Blind Reviews a joke?

fin.